What’s New in Cocoa

Session 115

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

What’s New in Cocoa

Ali Ozer and Kevin Perry
Cocoa Frameworks

What's New in Cocoa

Only on Mac OS

- We will:
  - Cover changes in Cocoa since Leopard
  - Give more coverage to changes since WWDC 2008
  - Provide pointers to other talks
- This is not "What's New in Cocoa Touch"

What's New in Cocoa Touch	Pacific Heights Tuesday 10:30AM

Topics

- 64-Bit
  - Replacements for Carbon APIs
- Blocks and Concurrency
- UTIs and Text Services
- NSCollectionView and Friends
- File System Efficiency
- And More
  - NSImage
  - Multi-Touch
  - Etc.

64-Bit

64-Bit

- In the works for a while
  - 10.4—System.framework
  - 10.5—All frameworks
  - 10.6—Almost all applications, commands, and daemons

"64-Bit Applications: Not in Leopard,
But Post-Leopard Likely Some Day."

WWDC 2006, Session 110: 64-Bit Overview

64-Bit Cocoa

- New types
  - NSInteger replaces int
  - NSUInteger replaces unsigned int
  - CGFloat replaces float

```
#if __LP64__
  typedef long          NSInteger;
  typedef unsigned long NSUInteger;
  typedef double        CGFloat;
#else
  typedef int           NSInteger;
  typedef unsigned int  NSUInteger;
  typedef float         CGFloat;
#endif
```

64-Bit Conversion

- Convert your sources to use the new types
- SnowLeopard has a new script to automate the process

```
/Developer/Extras/64BitConversion/ConvertCocoa64
```

- Run it on a copy of your sources

```
> cd MyAppSources
> /Developer/Extras/64BitConversion/ConvertCocoa64 `find . -name '*.[hm]' | xargs`
Processing ./AmazingView.m
Processing ./MyPlugIn/SomeSource.h
...
```

- Open original and new sources in FileMerge to check changes

64-Bit Conversion

- Check out the “64-Bit Transition Guide for Cocoa”
  - http://developer.apple.com/documentation/Cocoa/Conceptual/Cocoa64BitGuide/Introduction/Introduction.html

Carbon APIs in 64-Bit

- High-level Carbon frameworks not available
- Many replacement APIs have been added to Cocoa

NSApplication

- New APIs to control system user interface elements
  - Dock, menu bar, task switcher…

```
- (void)setPresentationOptions:(NSApplicationPresentationOptions)options;
- (NSApplicationPresentationOptions)presentationOptions;

- (NSApplicationPresentationOptions)currentSystemPresentationOptions;
```

- KVO-observable, even when another app changes the setting

NSApplication

- NSApplicationPresentationOptions

```
enum {
    NSApplicationPresentationDefault                    = 0,
    NSApplicationPresentationAutoHideDock               = (1 << 0),
    NSApplicationPresentationHideDock                   = (1 << 1),
    NSApplicationPresentationAutoHideMenuBar            = (1 << 2),
    NSApplicationPresentationHideMenuBar                = (1 << 3),
    NSApplicationPresentationDisableAppleMenu           = (1 << 4),
    NSApplicationPresentationDisableProcessSwitching    = (1 << 5),
    NSApplicationPresentationDisableForceQuit           = (1 << 6),
    NSApplicationPresentationDisableSessionTermination  = (1 << 7),
    NSApplicationPresentationDisableHideApplication     = (1 << 8),
    NSApplicationPresentationDisableMenuBarTransparency = (1 << 9)
};
```

NSWindow

- Disable server-side dragging

```
- (void)setMovable:(BOOL)flag;
- (BOOL)isMovable;
```

- Window “hit test”

```
+ (NSInteger)windowNumberAtPoint:(NSPoint)screenLocation
     belowWindowWithWindowNumber:(NSInteger)number;
```

- Collect window numbers of visible windows

```
+ (NSArray *)windowNumbersWithOptions:(NSWindowNumberListOptions)options;
```

NSRunningApplication

- Get info about applications running on the system

```
@interface NSRunningApplication : NSObject

@property (readonly, getter=isTerminated) BOOL         terminated;
@property (readonly, getter=isFinishedLaunching) BOOL finishedLaunching;
@property (readonly, getter=isHidden) BOOL             hidden;              // KVO
@property (readonly, getter=isActive) BOOL             active;              // KVO
@property (readonly) NSApplicationActivationPolicy     activationPolicy;    // KVO
@property (readonly) NSString                         *localizedName;
@property (readonly) NSString                         *bundleIdentifier;
@property (readonly) NSDate                           *launchDate;
@property (readonly) NSImage                          *icon;
@property (readonly) NSURL                            *bundleURL;
@property (readonly) NSURL                            *executableURL;
@property (readonly) NSInteger                         executableArchitecture;
@property (readonly) pid_t                             processIdentifier;
...
```

NSRunningApplication

- Hide, terminate, or activate applications

```
- (BOOL)hide;
- (BOOL)unhide;
- (BOOL)terminate;
- (BOOL)forceTerminate;
- (BOOL)activateWithOptions:(NSApplicationActivationOptions)options;
```

NSRunningApplication

- Getting a hold of running applications

```
+ (NSRunningApplication *)currentApplication;
+ (NSRunningApplication *)runningApplicationWithProcessIdentifier:(pid_t)pid;
+ (NSArray *)runningApplicationsWithBundleIdentifier:(NSString *)bundleIdentifier;
```

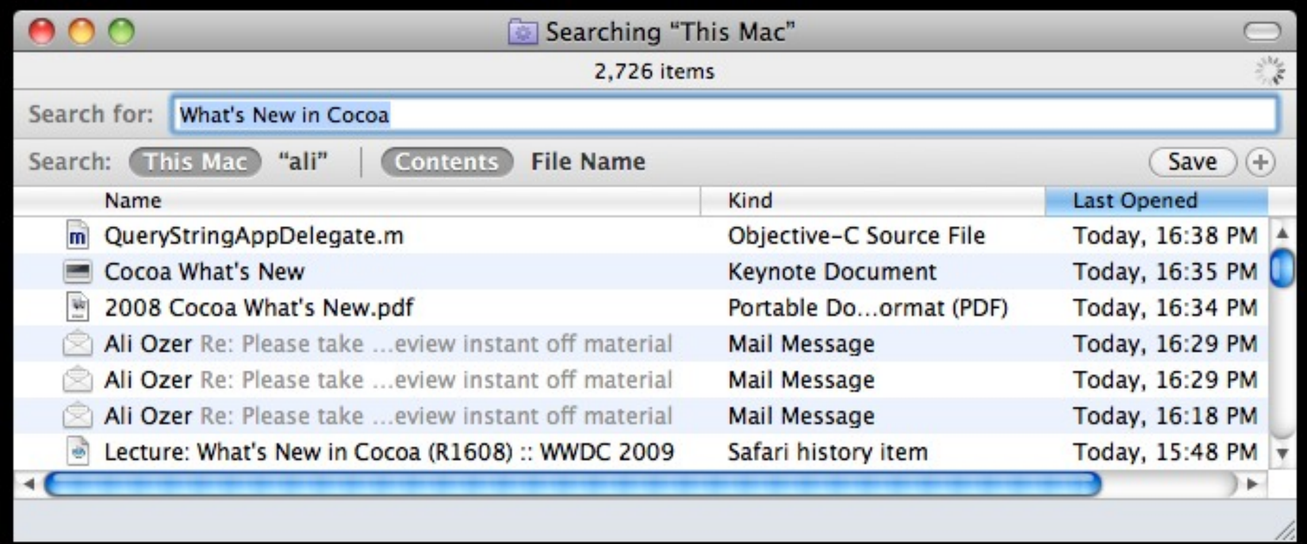
```
@interface NSWorkspace (NSWorkspaceRunningApplications)
- (NSArray *)runningApplications;
@end
```

NSWorkspace

- Display a Spotlight search results window in Finder for the specified query string

```
- (BOOL)showSearchResultsForQueryString:(NSString *)queryString;
```

- User can further fine tune the search using Finder

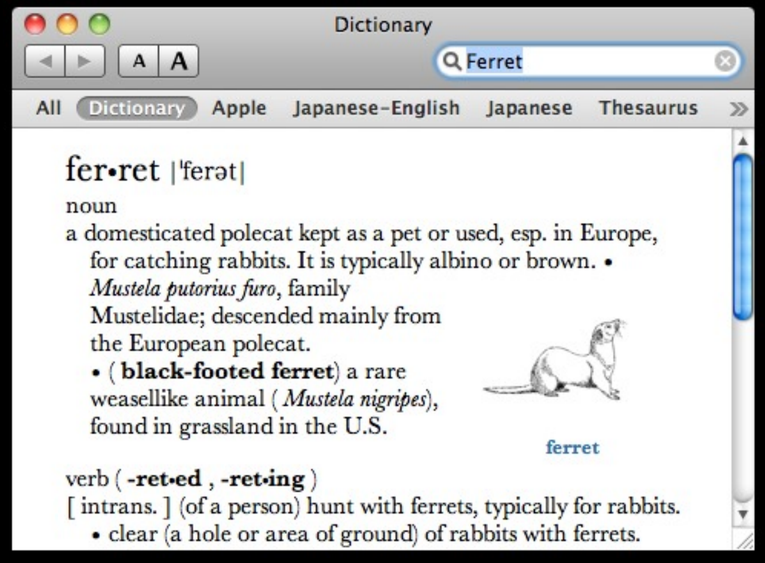


NSView

- Bring up Dictionary application for a piece of text

```
- (void)showDefinitionForAttributedString:(NSAttributedString *)attrString
                                  atPoint:(NSPoint)textBaselineOrigin;

- (void)showDefinitionForAttributedString:(NSAttributedString *)attrString
                                    range:(NSRange)targetRange
                                  options:(NSDictionary *)options
                    baselineOriginProvider:(NSPoint (^)(NSRange adjustedRange))provider;
```



NSPasteboard

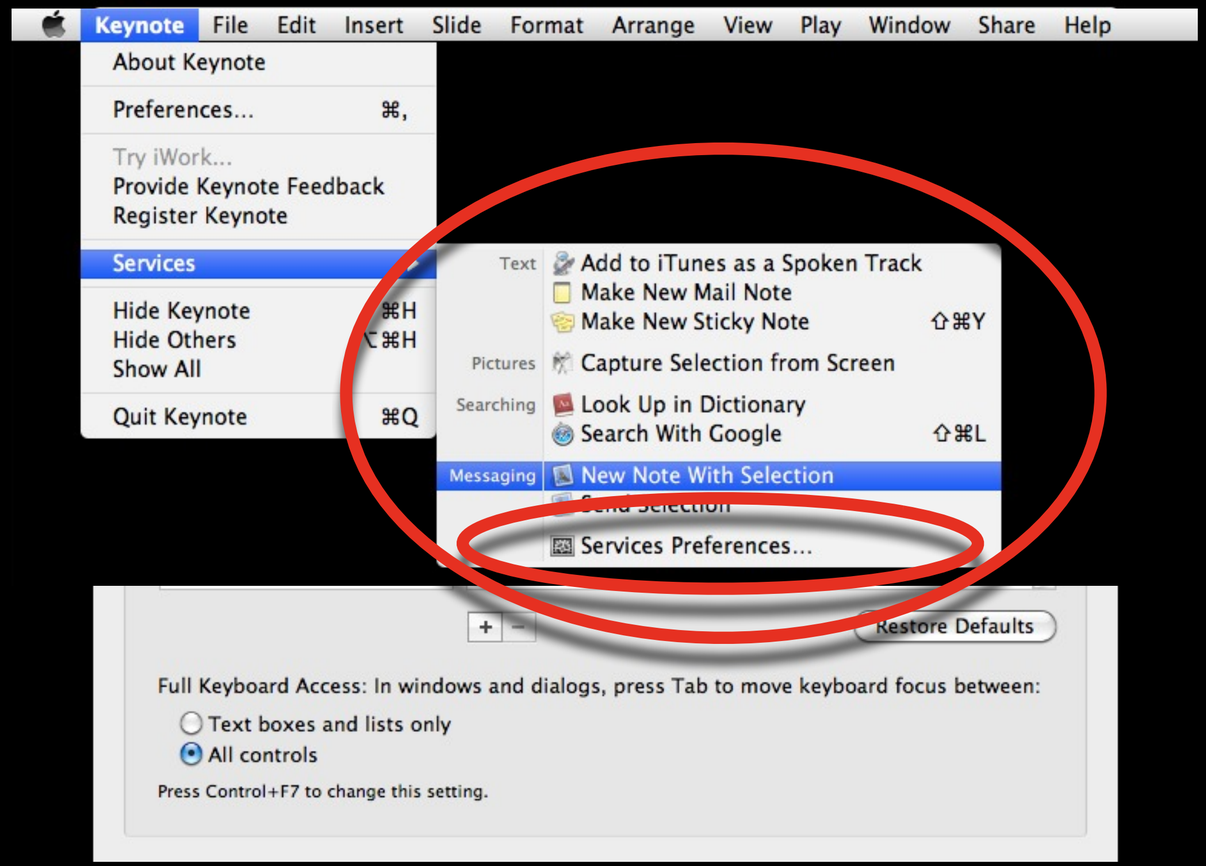
- Support for multiple items
- New, more general API
  - UTI-based
  - NSPasteboardWriting and NSPasteboardReading protocols
  - NSPasteboardItem class

Pasteboards, Services, and Interoperability on Mac

North Beach
Thursday 2:00PM

Services

- More discoverable
- Context sensitive
- User customizable



Pasteboards, Services, and Interoperability on Mac

North Beach
Thursday 2:00PM

NSMenu

- Display a menu as a popup

```
- (BOOL)popUpMenuPositioningItem:(NSMenuItem *)item
                      atLocation:(NSPoint)location
                          inView:(NSView *)view;
```

- Allow delegate to control menu location

```
- (NSRect)confinementRectForMenu:(NSMenu *)menu onScreen:(NSScreen *)screen;
```

- Stop menu tracking and dismiss menu

```
- (void)cancelTrackingWithoutAnimation;
```

NSEvent

- Get user preference settings

```
+ (NSTimeInterval)doubleClickInterval;
+ (NSTimeInterval)keyRepeatDelay;
+ (NSTimeInterval)keyRepeatInterval;
```

- Get keyboard and mouse state outside of the event stream

```
+ (NSUInteger)modifierFlags;
+ (NSUInteger)pressedMouseButtons;
```

NSEvent Monitoring

- Monitor (and affect) events in the current app

```
+ (id)addLocalMonitorForEventsMatchingMask:(NSEventMask)mask
                                   handler:(NSEvent *(^)(NSEvent *))block;
```

- Or monitor events sent to any app

```
+ (id)addGlobalMonitorForEventsMatchingMask:(NSEventMask)mask
                                    handler:(void (^)(NSEvent *))block;
```

- Stop monitoring

```
+ (void)removeMonitor:(id)eventMonitor;
```

User Events in Cocoa	Nob Hill Wednesday 3:30PM

Blocks

Blocks

- Snippets of code that capture lexical scope

```
int multiplier = 7;

int (^myBlock)(int) = ^(int num) { return num * multiplier; };
```

We're declaring a variable "myBlock." The "^" declares this to be a block

This is a literal block definition, assigned to variable myBlock

myBlock is a block that returns an int

It takes a single argument, also an int

The argument is named num

This is the body of the block

```
printf("%d", myBlock(3));          // Prints 21
```

Objective-C and Garbage Collection Advancements

Russian Hill
Wednesday 5:00PM

Blocks

- Used as an argument to an Objective-C method

Method Taking a Block Argument

Argument 1 (NSString)

Argument 2 (pointer to BOOL)

```
@interface NSString
- (void)enumerateLinesUsingBlock:(void (^)(NSString *line, BOOL *stop))block;
@end
```

Block Return Type (None here)

Argument Name

Text Processing in Cocoa

Presidio
Friday 9:00AM

Blocks

- Used in a new typedef

```
typedef NSComparisonResult (^NSComparator)(id obj1, id obj2);
```

Argument 1 (An Object)

Argument 2 (An Object)

Block Return Type

Type Being Defined

```
@interface NSArray
- (NSArray *)sortedArrayUsingComparator:(NSComparator)comparator;
@end
```

```
NSArray *sortedArray = [myArray sortedArrayUsingComparator:
            ^(id str1, id str2) {return [str1 localizedStandardCompare:str2];}
 ];
```

Block Naming

- “Block” is a challenging name
  - [myArray block];
  - [window beginSheet: block:]
- Instead of “block,” use more descriptive names
  - Handler
  - Provider
  - Test
  - Comparator
- “Block” only for the generic cases

Block-Based Collection Functionality

- NSArray sorting

```
- (NSArray *)sortedArrayUsingComparator:(NSComparator)comparator;
```

- Enumeration

```
- (void)enumerateObjectsUsingBlock:(void (^)(id obj, NSUInteger i, BOOL *stop))block;
```

- Searching

```
- (NSIndexSet *)indexesOfObjectsPassingTest:(BOOL (^)(id obj,
                                                      NSUInteger i,
                                                      BOOL *stop))predicate;
```

- Additional APIs
  - Versions of these for NSDictionary, NSSet, and NSIndexSet
  - Variants taking options and subset to work on

Concurrency

Concurrent Collection Functionality

- Enumerate, search, or sort concurrently
- Provide as option: NSEnumerationConcurrent or NSSortConcurrent

```
NSArray *array = [myArray sortedArrayWithOptions:NSSortConcurrent
                                 usingComparator:^(id str1, id str2) {
                                     return [str1 localizedStandardCompare:str2];
                                 }];
```

- The provided block may be invoked concurrently
- The overall operation is still synchronous
- The order of enumeration is undefined

NSOperation and NSOperationQueue

- Now implemented on top of GCD
- New API
  - NSOperation
    - Set thread priority
    - Specify a completion block
  - NSOperationQueue
    - Add multiple operations and optionally wait until all done
    - Add block as an operation
  - NSBlockOperation

NSNotificationCenter

- Now allows observers to get notifications concurrently

```
- (id)addObserverForName:(NSString *)name
                  object:(id)object
                   queue:(NSOperationQueue *)queue
              usingBlock:(void (^)(NSNotification *))block;
```

- Compare to Leopard API

```
- (void)addObserver:(id)observer
           selector:(SEL)aSelector
               name:(NSString *)name
             object:(id)object;
```

NSNotificationCenter

- Now allows observers to get notifications concurrently

```
- (id)addObserverForName:(NSString *)name
                  object:(id)object
                   queue:(NSOperationQueue *)queue
              usingBlock:(void (^)(NSNotification *))block;
```

- No change to the notification posting code
  - All observers finished before posting code continues

Concurrent Drawing

- NSView subclassers can opt-in to be drawn concurrently with other views

```
- (void)setCanDrawConcurrently:(BOOL)flag;
- (BOOL)canDrawConcurrently;
```

- -display or -displayIfNeeded blocked until all drawing done
  - In most cases thread-safety demands on the model unchanged

Concurrent Programming in Cocoa	North Beach Wednesday 9:00AM

TextEdit Enhancements

Concurrent Document Opening

Concurrent Document Opening

- Thread-safe document creation
  - init…
  - readFrom…
    - Disable undo registration
  - makeDocumentWithContentsOfURL:
  - openDocumentWithContentsOfURL:display:error:
    - TextEdit: Transient document

Concurrent Document Opening

- Override canConcurrentlyReadDocumentsOfType:

```
+ (BOOL)canConcurrentlyReadDocumentsOfType:(NSString *)typeName {
    NSWorkspace *workspace = [NSWorkspace sharedWorkspace];
    return !([workspace type:typeName
              conformsToType:(NSString *)kUTTypeHTML] ||
             [workspace type:typeName
              conformsToType:(NSString *)kUTTypeWebArchive]);
}
```

UTIs

- File type abstraction
  - Extensions
  - HFS type codes
  - MIME types
  - NSPasteboard type names
  - NSDocument type names
- Unified hierarchical type system
- Reduce complexity
- Snow Leopard recognizes over 400 UTIs out of the box
- Extensible

UTI Changes in Info.plist

```
<dict>
	<key>CFBundleTypeExtensions</key>
	<array>
		<string>rtf</string>
	</array>
	<key>CFBundleTypeIconFile</key>
	<string>rtf.icns</string>
	<key>CFBundleTypeMIMETypes</key>
	<array>
		<string>text/rtf</string>
	</array>
	<key>CFBundleTypeName</key>
	<string>NSRTFPboardType</string>
	<key>CFBundleTypeOSTypes</key>
	<array>
		<string>RTF </string>
	</array>
	<key>CFBundleTypeRole</key>
	<string>Editor</string>
	<key>LSIsAppleDefaultForType</key>
	<true/>
	<key>NSDocumentClass</key>
	<string>Document</string>
</dict>
```

UTI Adoption

- Info.plist
  - Use UTIs for Services declarations
- Switch from old type names to UTIs

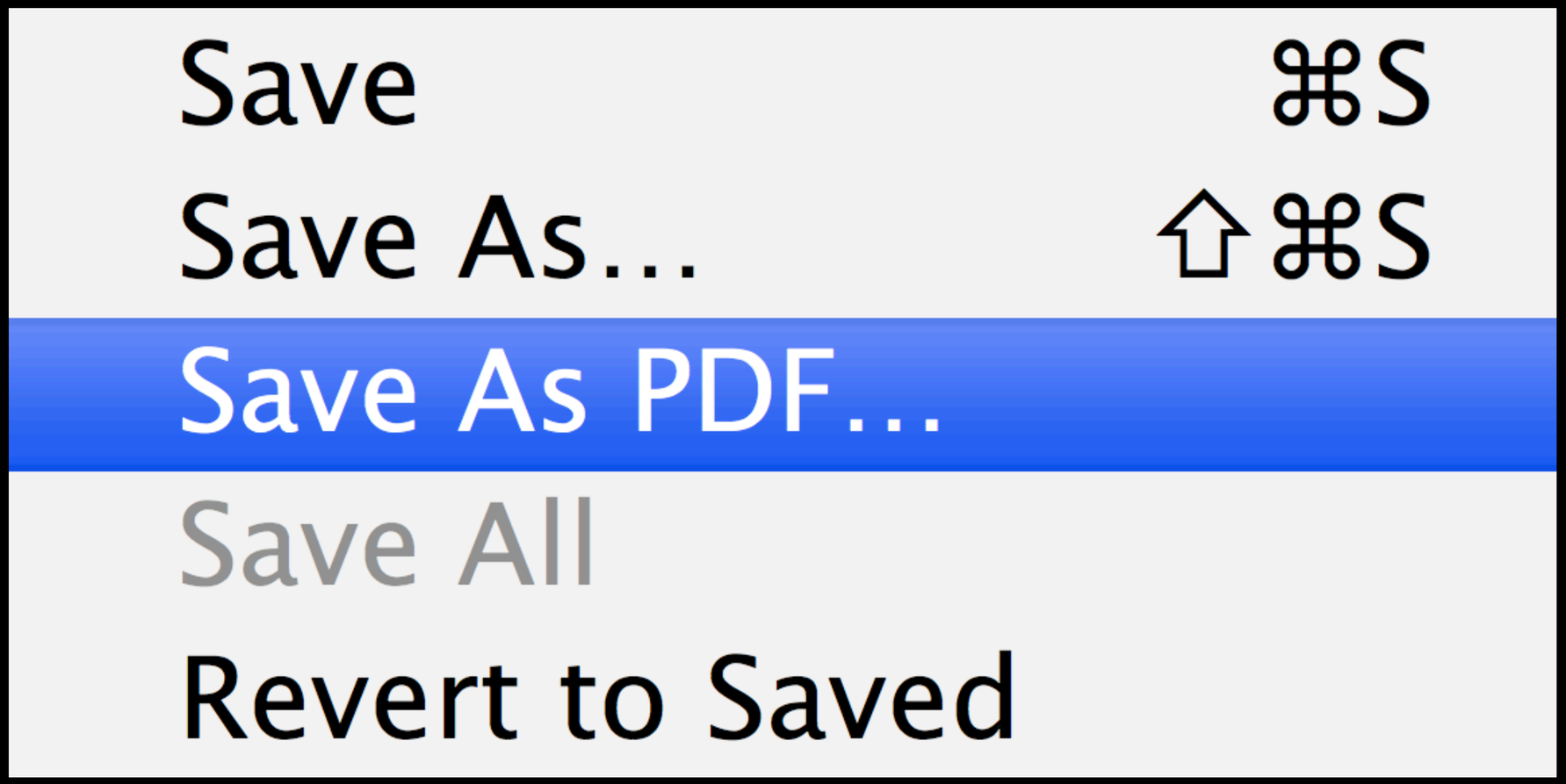
```
[aType isEqual:otherType];
```

```
[[NSWorkspace sharedWorkspace] type:aType conformsToType:otherType];
```

- Few changes

```
-[NSDocument readFromURL:ofType:error:]
-[NSDocument fileWrapperOfType:error:]
-[NSDocument writeableTypesForSaveOperation:]
-[NSDocument autosavingFileType]
-[NSDocumentController defaultType]
Services methods
```

Save as PDF…



Save as PDF…

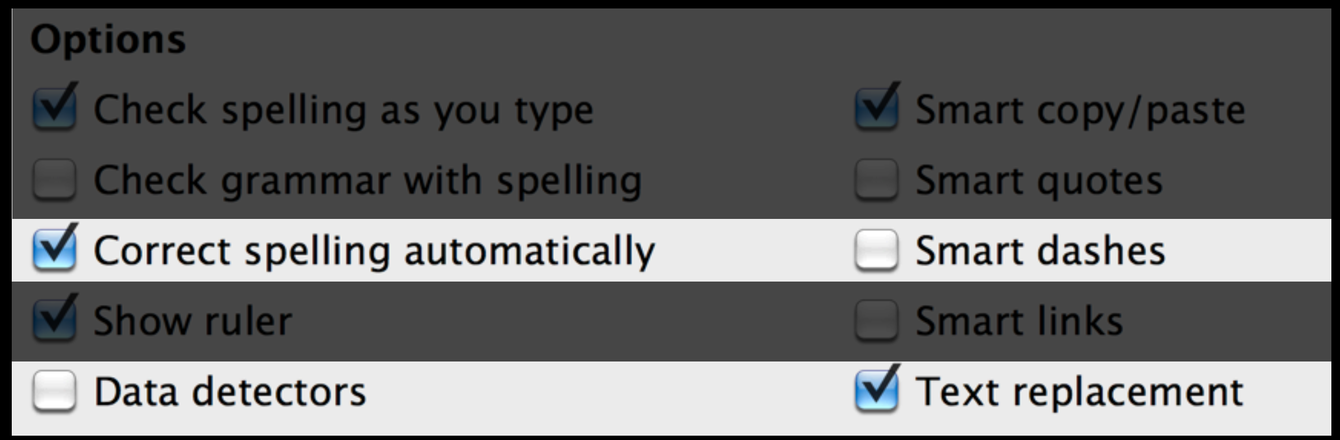
```
- (IBAction)saveDocumentAsPDFTo:(id)sender {
    NSSavePanel *savePanel = [NSSavePanel savePanel];
    NSString *path = [[self fileURL] path];
    NSString *defaultPDFFileName = [[path lastPathComponent] stringByDeletingPathExtension];
    [savePanel setRequiredFileType:@"com.adobe.pdf"];
    [savePanel setCanSelectHiddenExtension:YES];
    if (defaultPDFFileName) [savePanel setNameFieldStringValue:defaultPDFFileName];
    NSWindow *window = [[[self windowControllers] objectAtIndex:0] window];
    [savePanel beginSheetModalForWindow:window completionHandler:^(NSInteger result) {
        if (result == NSFileHandlingPanelOKButton) {
            NSDictionary *settings = [NSDictionary dictionaryWithObjectsAndKeys:
                NSPrintSaveJob, NSPrintJobDisposition,
                [savePanel filename], NSPrintSavePath,
                nil];
            [self printDocumentWithSettings:settings
                               showPrintPanel:NO
                                     delegate:nil
                             didPrintSelector:NULL
                                  contextInfo:NULL];
            if ([savePanel isExtensionHidden]) {
                NSDictionary *attributes = [NSDictionary dictionaryWithObjectsAndKeys:[NSNumber numberWithBool:YES],
NSFileExtensionHidden, nil];
                [[NSFileManager defaultManager] setAttributes:attributes ofItemAtPath:[savePanel filename] error:NULL];
            }
        }
    }];
}
```

Text Checking

- Spell checking
- Leopard
  - Grammar checking
  - Quote substitution
  - Link detection
- Snow Leopard
  - Language and script identification
  - Data detectors
  - Dash substitution
  - Automatic text replacement and spelling correction

Text Checking in TextEdit

- Preferences



- Configure NSTextView

```
-setAutomaticSpellingCorrectionEnabled:
-setAutomaticDashSubstitutionEnabled:
-setAutomaticDataDetectionEnabled:
-setAutomaticTextReplacementEnabled:
```

Text Checking

- New APIs
  - NSTextView
  - NSSpellChecker

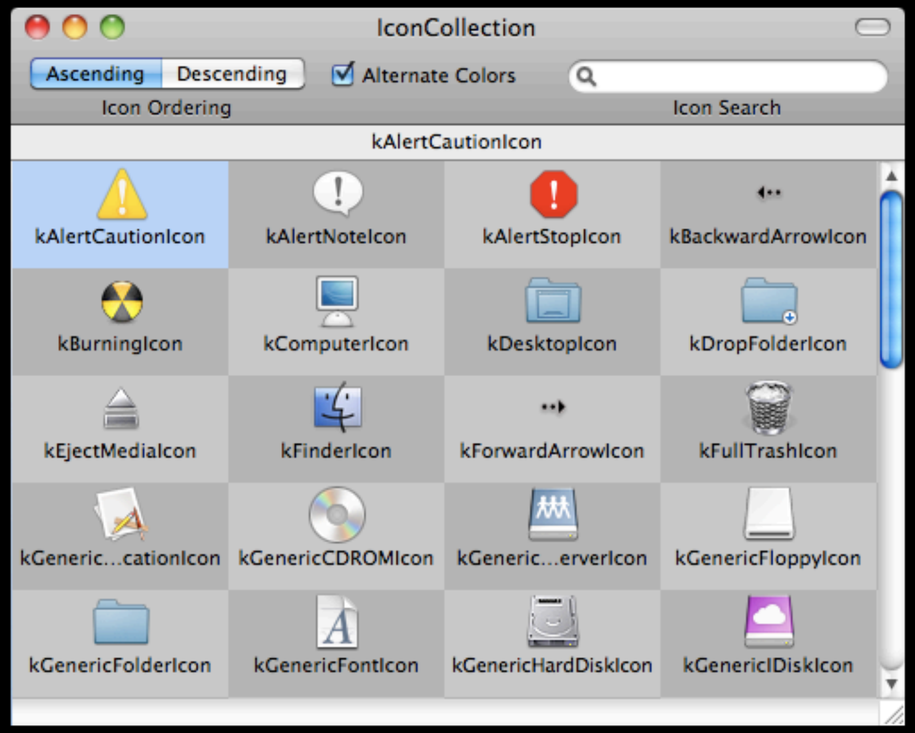
Text Processing in Cocoa	Presidio Friday 9:00AM

TextEdit Enhancements

- Concurrent document opening
  - Thread-safe document creation + one simple method override
- UTI adoption
- Save as PDF
  - UI + one simple method
- Text checking
  - UI + prefs + four method calls

NSCollectionView

NSCollectionView



- Introduced in Leopard
- Lots of improvements in Snow Leopard

Bug Fixes!

- Smoother relayout animations
- More predictable layout
- Autoresizing
- KVO-compliant -selectionIndexes
- More reliable
- Much more!

Nib-Based Items

- Binding to NSCollectionViewItem

```
@interface NSCollectionViewItem : NSViewController
```

  - Move prototype to dedicated nib
  - Connect prototype view and item (file's owner)
    - Set nib name and bundle identifier
- Exact replication
- Efficient, lazy loading
- Backwards compatible

Demo

Drag and Drop

- Familiar delegate APIs
  - Use indexes for efficiency
  - New APIs

```
- (NSCollectionViewItem *)itemAtIndex:(NSUInteger)index;
- (NSRect)frameForItemAtIndex:(NSUInteger)index;
```

- Local reordering
- Compatible with all NSDragPasteboard clients
- Click and hold

NSTableView

- Reload specific rows and columns
- Selection highlight
- Drag destination feedback
- Control focusing on cells or columns
- Configure column size-to-fit behavior
- Control column reordering

Presenting User Data with Table Views and Browsers	Nob Hill Thursday 10:30AM

NSBrowser

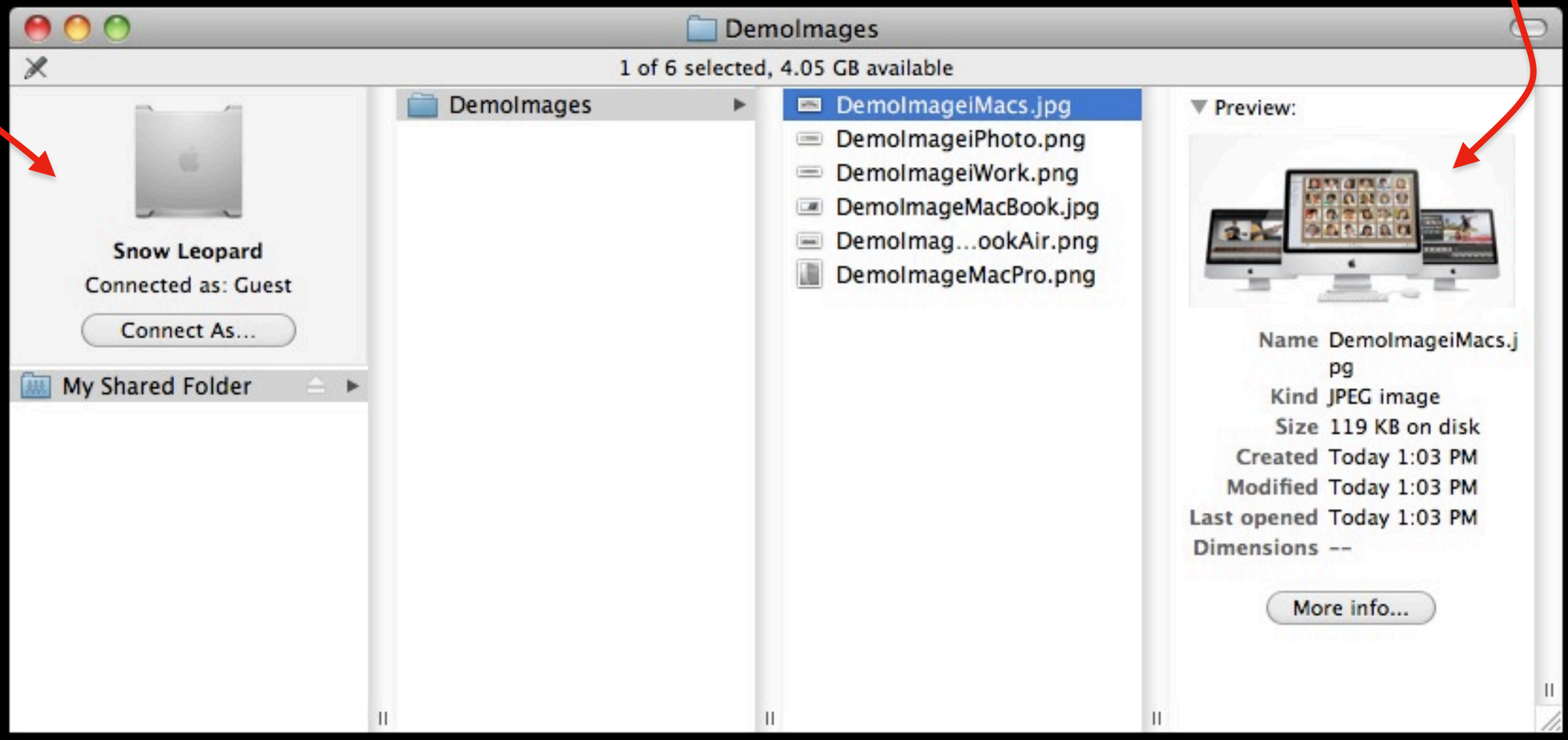
- Item-based delegate APIs
- Most new APIs require item-based delegate
- Variable row heights
- Editing
- Custom preview columns and column headers

Presenting User Data with Table Views and Browsers	Nob Hill Thursday 10:30AM

NSBrowser

Column header

Preview column



File System Efficiency

NSURL

NSURL

- Much more complete access to information about files
  - Eliminates need to convert between POSIX paths, FSRefs, NSStrings
- Caching to avoid hitting the disk
- Better error reporting

NSURL Property Access

```
- (BOOL)getResourceValue:(id *)value forKey:(NSString *)key error:(NSError **)error;
- (BOOL)setResourceValue:(id)value   forKey:(NSString *)key error:(NSError **)error;

- (NSDictionary *)resourceValuesForKeys:(NSArray *)keys error:(NSError **)error;
- (BOOL)setResourceValues:(NSDictionary *)keyedValues   error:(NSError **)error
```

```
NSString *NSURLNameKey;
NSString *NSURLLocalizedNameKey;
NSString *NSURLIsPackageKey;
NSString *NSURLHasHiddenExtensionKey;
NSString *NSURLCreationDateKey;         // NSDate
NSString *NSURLVolumeURLKey;
NSString *NSURLCustomIconKey;           // NSImage
NSString *NSURLEffectiveIconKey;        // NSImage
NSString *NSURLLabelColorKey;           // NSColor
NSString *NSURLFileSizeKey;
...
NSString *NSURLVolumeAvailableCapacityKey;
...
```

NSURL Path Utilities

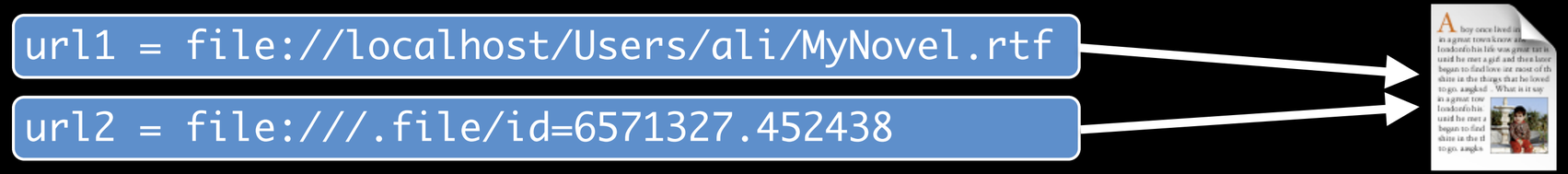
```
+ (NSURL *)fileURLWithPathComponents:(NSArray *)components;
- (NSArray *)pathComponents;
- (NSString *)lastPathComponent;
- (NSString *)pathExtension;
- (NSURL *)URLByAppendingPathComponent:(NSString *)pathComponent;
- (NSURL *)URLByDeletingLastPathComponent;
- (NSURL *)URLByAppendingPathExtension:(NSString *)pathExtension;
- (NSURL *)URLByDeletingPathExtension;

- (NSURL *)URLByStandardizingPath;
- (NSURL *)URLByResolvingSymlinksInPath;
```

File Reference URL

- Tracks a file by ID, so that it's referenced even after a rename
- Create one with -[NSURL fileReferenceURL]

```
NSURL *url1 = [NSURL fileURLWithPath:@"/Users/ali/MyNovel.rtf"];
NSURL *url2 = [url1 fileReferenceURL];
```



- Mostly transparent
- For specialized uses
- Not meant for persistence

NSURL Bookmarks

- NSData containing persistent form of an NSURL
- To create a bookmark

```
- (NSData *)bookmarkDataWithOptions:(NSURLBookmarkCreationOptions)options
     includingResourceValuesForKeys:(NSArray *)keys
                      relativeToURL:(NSURL *)relativeURL
                              error:(NSError **)error;
```

- To recreate NSURL from a bookmark

```
+ (id)URLByResolvingBookmarkData:(NSData *)bookmarkData
                         options:(NSURLBookmarkResolutionOptions)options
                   relativeToURL:(NSURL *)relativeURL
             bookmarkDataIsStale:(BOOL *)isStale
                           error:(NSError **)error;
```

NSFileManager

- NSURL versions of existing functionality; for example

```
- (BOOL)copyItemAtURL:(NSURL *)srcURL toURL:(NSURL *)dstURL error:(NSError **)error;
- (BOOL)moveItemAtURL:(NSURL *)srcURL toURL:(NSURL *)dstURL error:(NSError **)error;
- (BOOL)removeItemAtURL:(NSURL *)URL error:(NSError **)error;
...
```

- Directory enumeration

```
- (NSDirectoryEnumerator *)enumeratorAtURL:(NSURL *)url
                includingPropertiesForKeys:(NSArray *)keys
                                   options:(NSDirectoryEnumerationOptions)mask
                              errorHandler:(BOOL (^)(NSURL *url, NSError *err))hdlr;
```

  - Keys argument is a hint
  - Options allow skipping subdirectories, package contents, hidden files
  - A “NO” return from error handler stops enumeration

NSFileManager Safe Saving

```
- (BOOL)replaceItemAtURL:(NSURL *)originalItemURL
           withItemAtURL:(NSURL *)newItemURL
          backupItemName:(NSString *)backupItemName
                 options:(NSFileManagerItemReplacementOptions)options
        resultingItemURL:(NSURL **)resultingURL
                   error:(NSError **)error;
```

- NSDocument uses this
- You may never need to

NSFileWrapper

- API clean up
  - NSURL
  - NSError
  - Control over lazy and mapping behaviors
- More complete support for metadata

NSFileWrapper Reading

```
- (id)initWithURL:(NSURL *)url
          options:(NSFileWrapperReadingOptions)opts
            error:(NSError **)error;

- (BOOL)readFromURL:(NSURL *)url
            options:(NSFileWrapperReadingOptions)opts
              error:(NSError **)error;
```

```
enum {
    NSFileWrapperReadingImmediate         = 1,
    NSFileWrapperReadingWithoutMapping    = 2
};
```

- NSFileWrapper is normally lazy and may use mapping
  - Reading options allow turning these off

NSFileWrapper Writing

```
- (BOOL)writeToURL:(NSURL *)url
           options:(NSFileWrapperWritingOptions)options
originalContentsURL:(NSURL *)original
             error:(NSError **)error;
```

- Enables incremental saving
  - Provide previous version of document as originalContentsURL

NSWorkspace

- Asynchronous file copy and recycle

```
- (void)recycleURLs:(NSArray *)URLs
          completionHandler:(void (^)(NSDictionary *newURLs, NSError *error))handler;
- (void)duplicateURLs:(NSArray *)URLs
          completionHandler:(void (^)(NSDictionary *newURLs, NSError *error))handler;
```

- Example: Attempt to copy three files
  - One already copied, one not, and one nonexistent

```
newURLs = {
    "file://localhost/tmp/NotYetCopied" = "file://localhost/tmp/NotYetCopied copy";
    "file://localhost/tmp/AlreadyCopied" = "file://localhost/tmp/AlreadyCopied copy 2";
}
error = The file “Nonexistent” could not be duplicated because it was not found.
```

And More…

NSSavePanel

- Block-based sheet APIs

```
- (void)beginWithCompletionHandler:(void (^)(NSInteger result))handler;

- (void)beginSheetModalForWindow:(NSWindow *)window
               completionHandler:(void (^)(NSInteger result))handler;
```

NSSavePanel Block APIs

Bringing up a sheet in Leopard and Snow Leopard

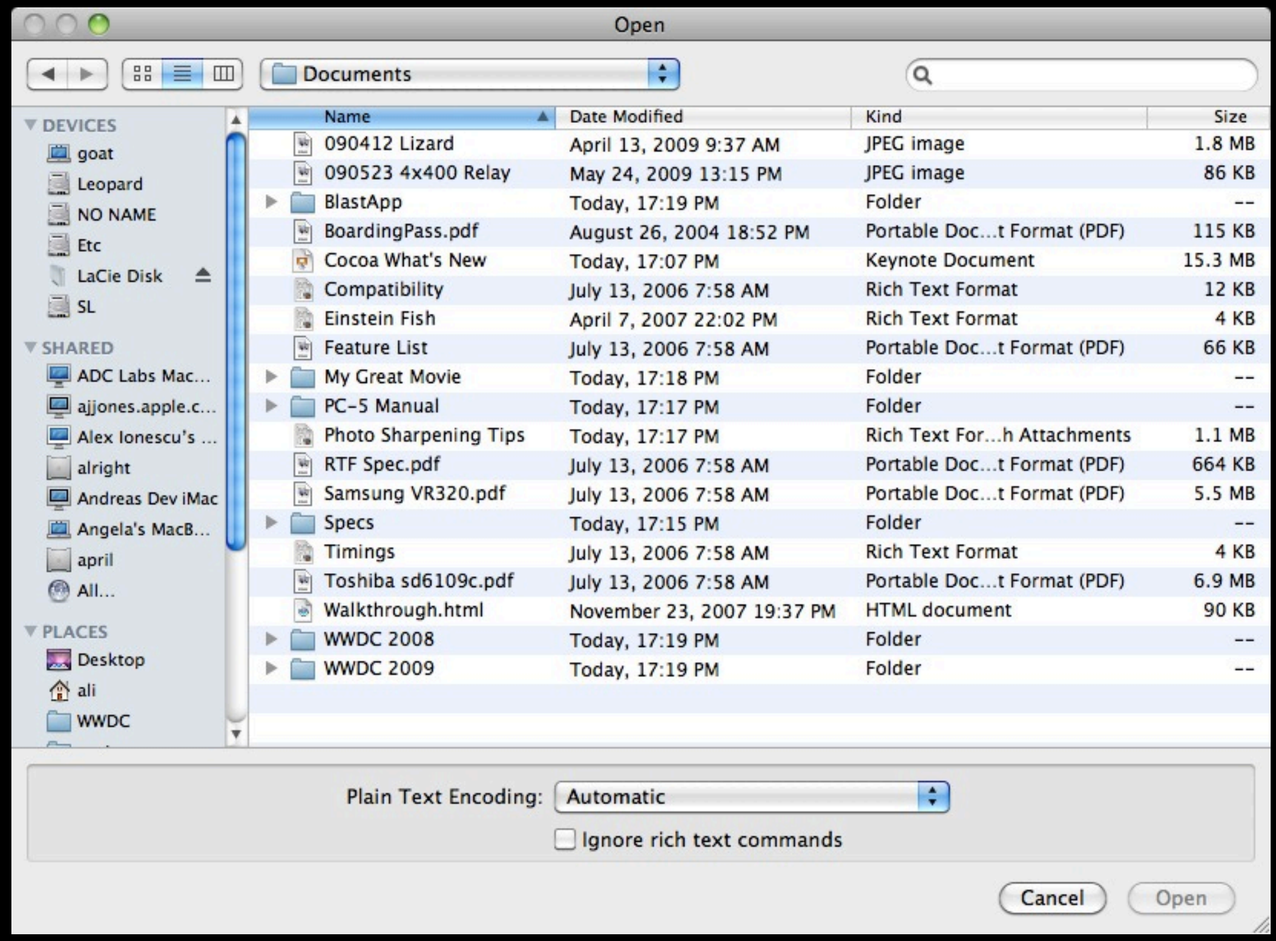
```
- (void)saveInLeopard:(id)sender {
    NSSavePanel *panel = [NSSavePanel savePanel];
    [panel beginSheetForDirectory:nil file:@"" modalForWindow:window
                    modalDelegate:self
                   didEndSelector:@selector(saveDone:returnCode:context:)
                      contextInfo:NULL];
}

- (void)saveDone:(NSSavePanel *)panel returnCode:(NSInteger)result context:(void *)ctxt {
    if (result == NSFileHandlingPanelOKButton) NSLog(@"File: %@", [panel URL]);
}
```

```
- (void)saveInSnowLeopard:(id)sender {
    NSSavePanel *panel = [NSSavePanel savePanel];
    [panel beginSheetModalForWindow:window completionHandler:^(NSInteger result) {
        if (result == NSFileHandlingPanelOKButton) NSLog(@"File: %@", [panel URL]);
    }];
}
```

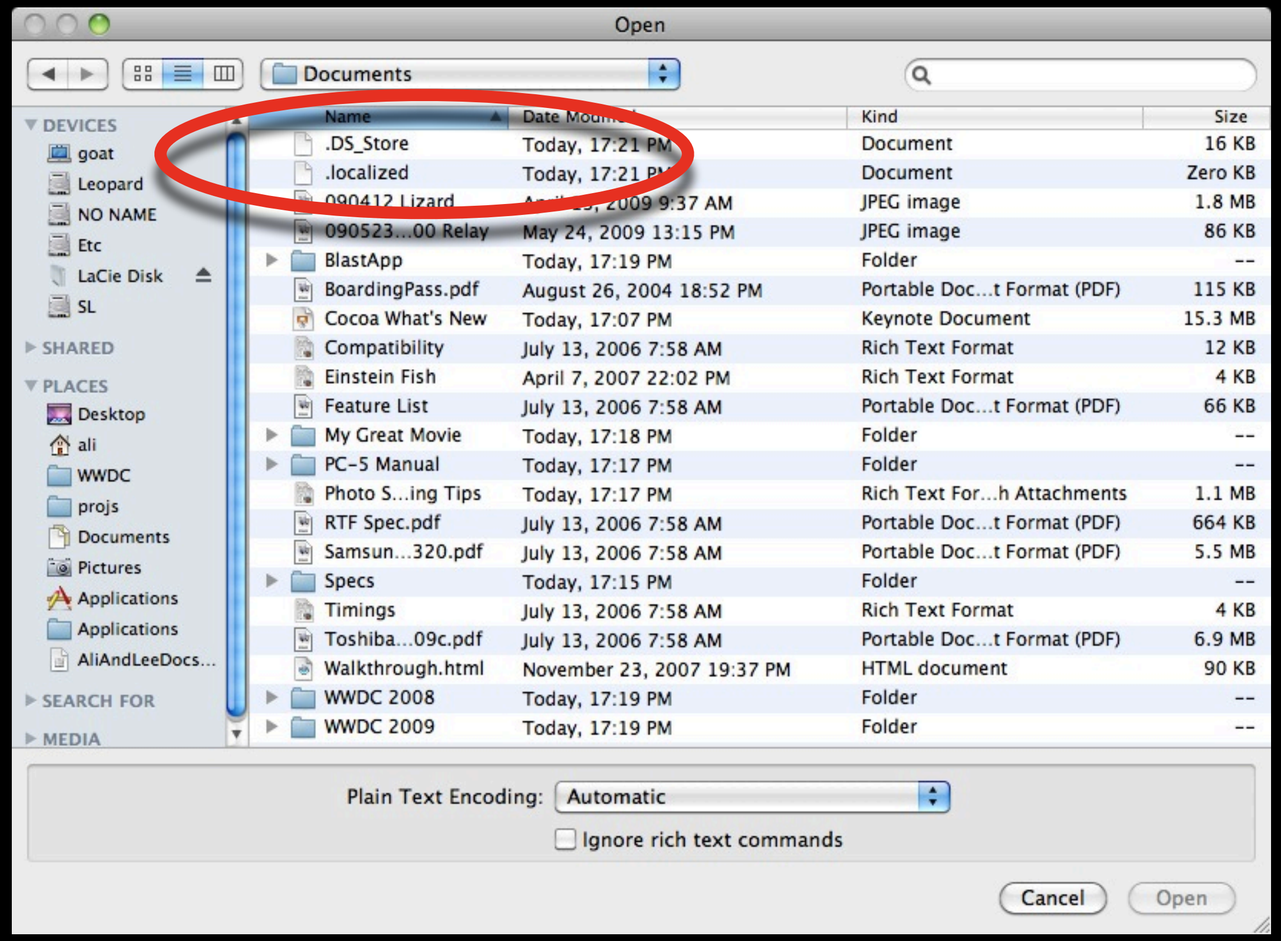
Open/Save Panel Tidbits

- Right click on the column header in list view to add additional columns



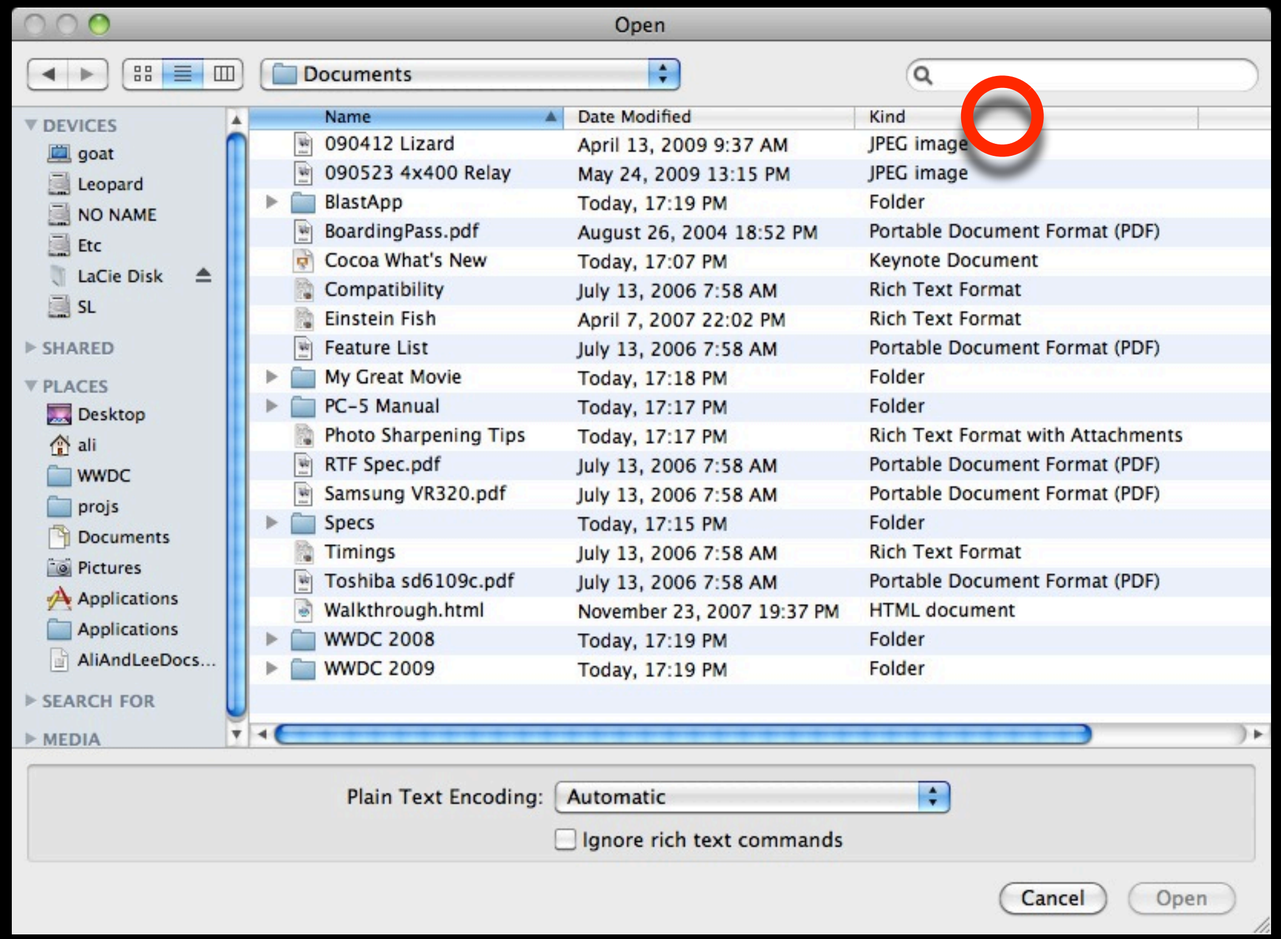
Open/Save Panel Tidbits

- Hit cmd-shift-period to temporarily view hidden files



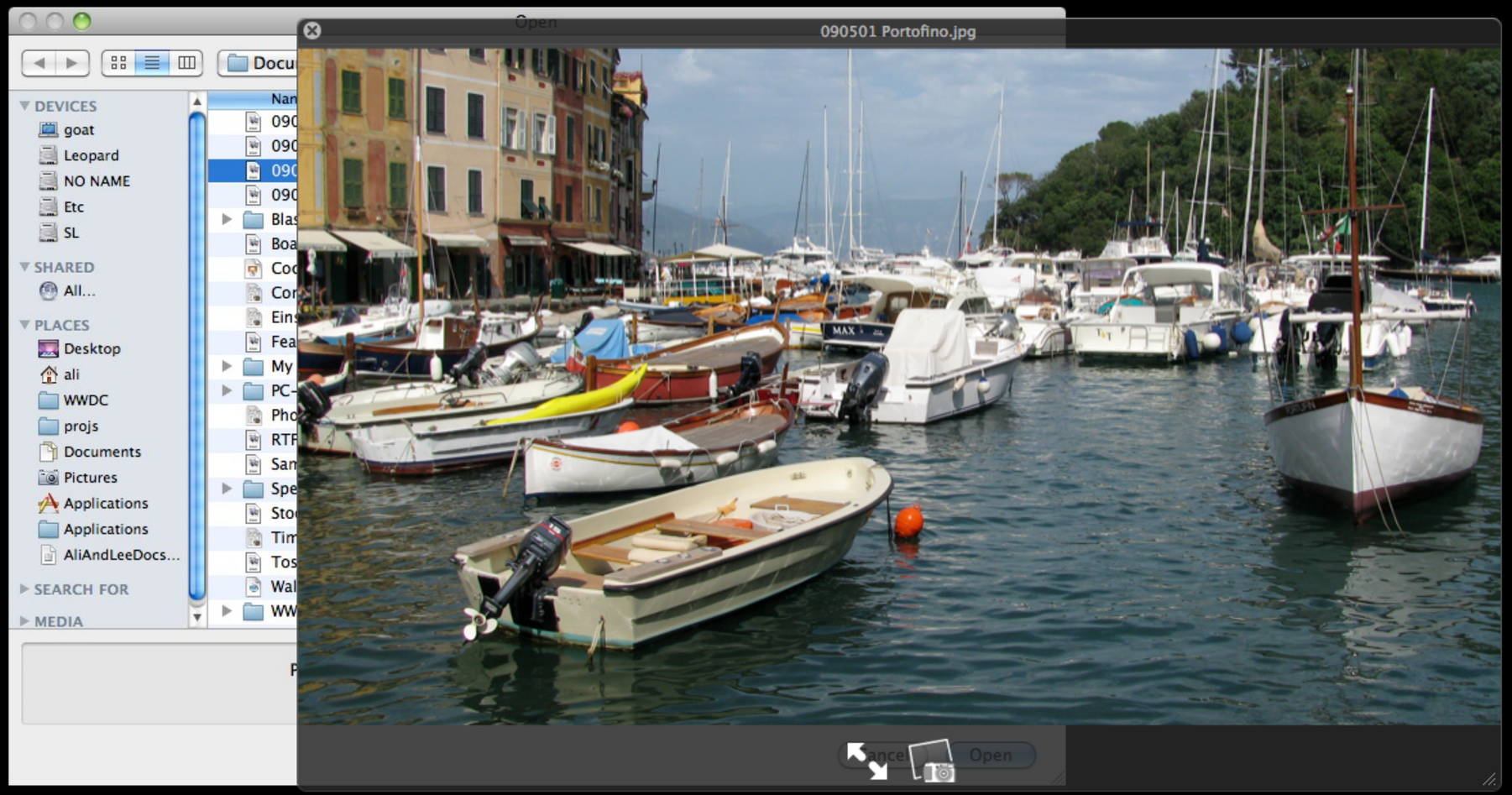
Open/Save Panel Tidbits

- Double-click right edge of header to automatically resize a column



Open Panel Tidbits

- Hit space in open panel to Quick Look selected files



NSImage

- Improved performance
  - Better "impedance match" with CGImage
  - No more caching of images in windows
- Some API changes as well

New in NSImage

- Ability to create with and extract a CGImage
- Better control over drawing parameters
- Hit-test a point or a rectangle
- Automatic interpretation of orientation info
- Ability to set accessibility description
- Ability to pass NSImage into -[CALayer setContents:]
- Many new standard images

NSImage in Snow Leopard	Nob Hill Thursday 3:30PM

NSView Layer Redrawing

- Specify when NSView layers are redrawn

```
- (void)setLayerContentsRedrawPolicy:(NSViewLayerContentsRedrawPolicy)policy;
- (NSViewLayerContentsRedrawPolicy)layerContentsRedrawPolicy;
```

- NSViewLayerContentsRedrawPolicy

```
enum {
    NSViewLayerContentsRedrawNever                  = 0,
    NSViewLayerContentsRedrawOnSetNeedsDisplay      = 1,
    NSViewLayerContentsRedrawDuringViewResize       = 2,
    NSViewLayerContentsRedrawBeforeViewResize       = 3
};
```

NSView Layer Redrawing

- Specify what to do when the layers are not redrawn

```
- (void)setLayerContentsPlacement:(NSViewLayerContentsPlacement)newPlacement;
- (NSViewLayerContentsPlacement)layerContentsPlacement;
```

- NSViewLayerContentsPlacement

```
NSViewLayerContentsPlacementScaleAxesIndependently      =  0,
NSViewLayerContentsPlacementScaleProportionallyToFit    =  1,
NSViewLayerContentsPlacementScaleProportionallyToFill   =  2,
NSViewLayerContentsPlacementCenter                      =  3,
NSViewLayerContentsPlacementTop                         =  4,
NSViewLayerContentsPlacementTopRight                    =  5,
NSViewLayerContentsPlacementRight                       =  6,
NSViewLayerContentsPlacementBottomRight                 =  7,
NSViewLayerContentsPlacementBottom                      =  8,
NSViewLayerContentsPlacementBottomLeft                  =  9,
NSViewLayerContentsPlacementLeft                        = 10,
NSViewLayerContentsPlacementTopLeft                     = 11
```

Multi-Touch

- Support for standard gestures on the trackpads of recent notebooks
  - Magnify
  - Swipe
  - Rotate
- Also support for arbitrary touch events

Multi-Touch

- Opt-in at the NSView level

```
- (void)setAcceptsTouchEvents:(BOOL)flag;
- (BOOL)acceptsTouchEvents;
```

- New NSResponder methods

```
- (void)touchesBeganWithEvent:(NSEvent *)event;
- (void)touchesMovedWithEvent:(NSEvent *)event;
- (void)touchesEndedWithEvent:(NSEvent *)event;
- (void)touchesCancelledWithEvent:(NSEvent *)event;
```

- New class NSTouch

User Events in Cocoa	Nob Hill Wednesday 3:30PM

NSDockTilePlugIn

- Enables an application to customize its dock tile when not running

```
@protocol NSDockTilePlugIn <NSObject>

@required
- (void)setDockTile:(NSDockTile *)dockTile;

@optional
- (NSMenu*)dockMenu;

@end
```

- Create a plug-in inside the application bundle
- Declare it via NSDockTilePlugIn in Info.plist
- Implement NSDockTilePlugIn on your principal class

Sudden Termination

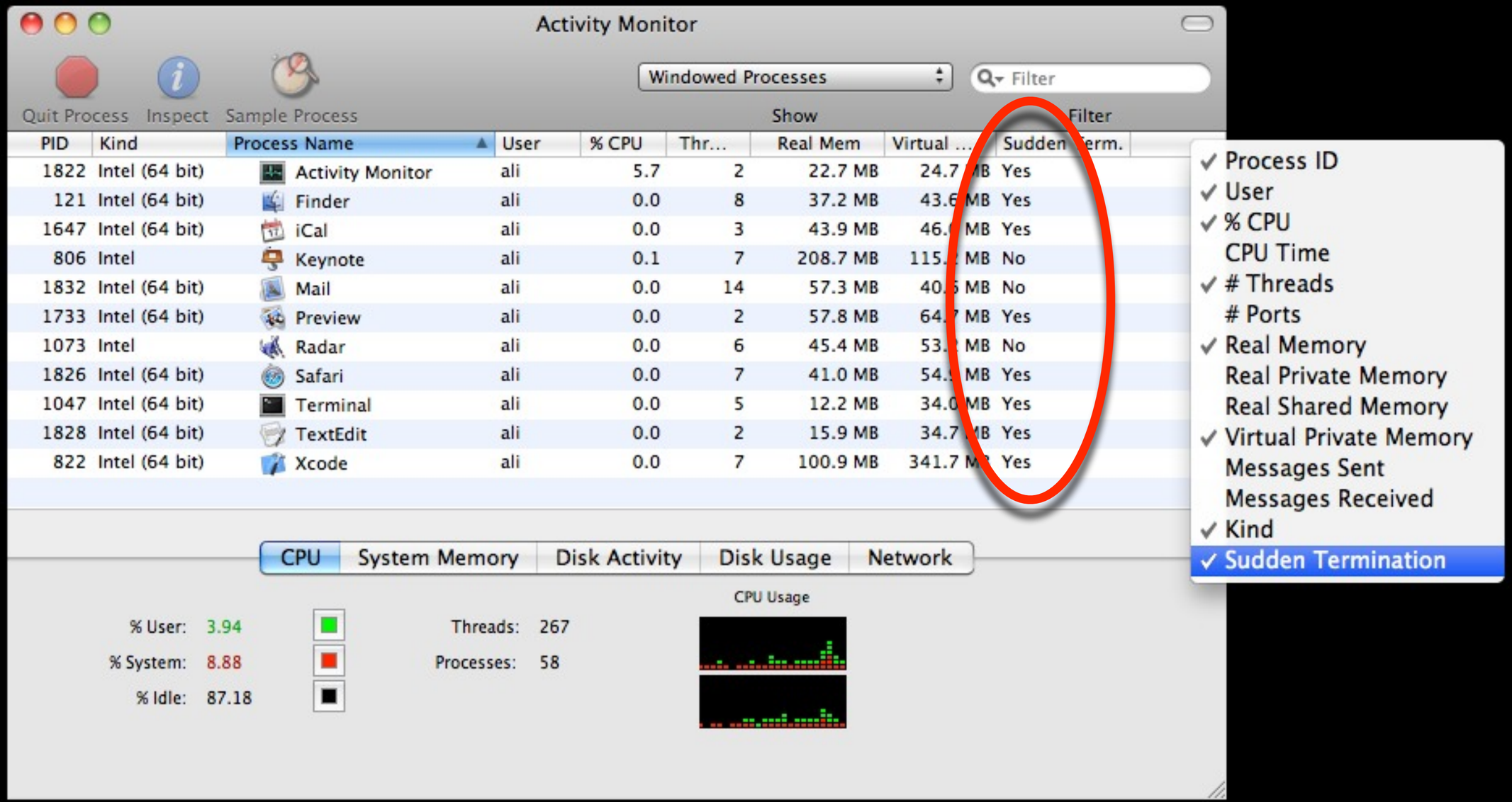
- Enables “clean” apps to be quit very quickly
- API to mark regions where the app isn’t clean

```
@interface NSProcessInfo
- (void)disableSuddenTermination;
- (void)enableSuddenTermination;
@end
```

- AppKit marks app as dirty during event dispatching
- Requires opt-in

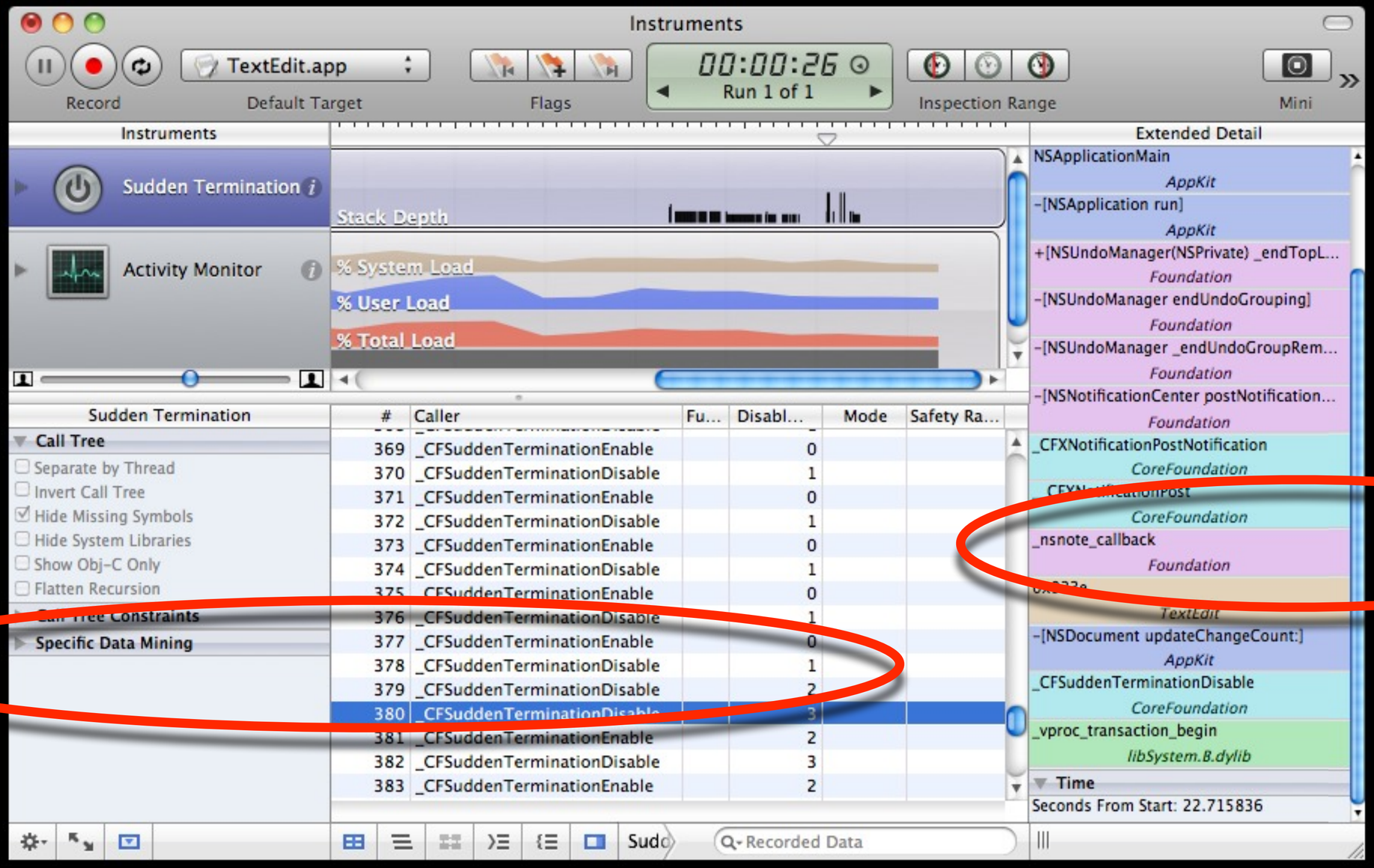
Sudden Termination

Activity Monitor support



Sudden Termination

Instruments support



Other Important Changes

- Formal protocol declarations for delegates and data sources
- New classes NSCache and NSPurgeableData
- NSStream support for property lists
- NSWindow and NSScreen color space support
- NSWindow display throttling (to beam-sync)
- Selection printing
- NSUserInterfaceItemSearching protocol
- New class NSTextInputContext
- Improvements in editing of bidirectional text
- Accessibility improvements

For More Information

- Release notes
  - AppKit (“Cocoa Application Framework”)
  - Foundation (“Cocoa Foundation Framework”)
- Changes docs
- Header files
  - Look for “10_6”
    - AVAILABLE_MAC_OS_X_VERSION_10_6_AND_LATER
    - DEPRECATED_IN_MAC_OS_X_VERSION_10_6_AND_LATER
    - MAC_OS_X_VERSION_10_6
- Documentation

A Few More Things You Can Do in 10.6

Read the release notes to find out more!

- Put URLs directly in preferences
- Sort file names “like Finder”
- Change the desktop image
- Get compiler warning for “NSLog(message)”
- Have NSErrors display a help button in alerts
- Animate integer and color-valued properties in NSView
- Have your finalized KVO observers automatically removed under GC

Related Sessions

State of the Art Cocoa: Application Design	Nob Hill Tuesday 10:30AM
State of the Art Cocoa: User Interface Design	Nob Hill Tuesday 2:00PM
Designing Your Cocoa Application for Concurrency	Nob Hill Tuesday 3:30PM
Concurrent Programming in Cocoa	North Beach Wednesday 9:00AM
What's New in Core Data	North Beach Wednesday 10:30AM
User Events in Cocoa	Nob Hill Wednesday 3:30PM
Objective-C and Garbage Collection Advancements	Russian Hill Wednesday 5:00PM

Related Sessions

Presenting User Data with Table Views and Browsers	Nob Hill Thursday 10:30AM
Pasteboards, Services, and Interoperability on Mac	North Beach Thursday 2:00PM
NSImage in Snow Leopard	Nob Hill Thursday 3:30PM
Cocoa Tips and Tricks	Nob Hill Thursday 5:00PM
Text Processing in Cocoa	Presidio Friday 9:00AM
Advanced Accessibility: Best Practices and New APIs	Mission Friday 2:00PM

Labs

Cocoa Open Lab	OS X Lab C Tuesday 3:30–6:15PM
Concurrent Programming in Cocoa Lab	OS X Lab D Thursday 9:00–11:45AM
Cocoa Open Lab	OS X Lab C Thursday 2:00–4:45PM
Core Data Lab	Snow Leopard Lab Friday 9:00–11:45AM

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09